



PRODUCTIE DOSSIER

Owen De Waele NMD | 2019 - 20

Bachelor in de grafische en digitale media |
Bachelor of Graphical and Digital Media

Anteveld Hogeschool | Anteveld University College Ghent

Inhoudsopgave

Briefing	3
Analyse.....	6
Planning.....	8
Sitemap	10
Wireframes	12
Wireflow	14
Designs.....	16
Style guide.....	18
Code snippets.....	20
Userguide	24
Deploymentguide.....	29

Briefing

Introductie

Jouw klant ARTE-TECH, een technisch onderhoudsbedrijf komt met de vraag om de opvolging van hun werken en opvolging van het personeel, aannemers, onderaannemers, ... te optimaliseren.

Activiteiten van de klant

ARTE-TECH biedt de dienst aan om technisch geschoolde mensen gedurende een bepaalde periode ter plaatse naar bedrijven te sturen om daar gevraagde onderhoudswerken uit te voeren. Zowel tijdens een shutdown als voor het dagelijks onderhoud zorgen ze ervoor dat de juiste mensen snel op locatie zijn om assistentie te bieden of volledig autonoom een machine te repareren.

Probleemstelling

ARTE-TECH beschikt enerzijds over eigen personeel, maar beschikt anderzijds ook over een aantal onderaannemers (vergelijkbaar met freelancers) die ze - in naam van ARTE-TECH - kunnen uitsturen naar bedrijven. De opvolging hiervan is moeilijk want iedereen heeft een andere manier van werken.

De freelancers werken met eigen werkbonnen (startuur, einduur, activiteiten van de dag, verplaatsingen, ...).

Personeel houdt dan weer alle uren en activiteiten op papier bij en laat die wekelijks ondertekenen door hun klant (het bedrijf waar ze onderhoud doen).

Elke week verzamelt de manager van ARTE-TECH alle paperassen, werkbonnen en e-mails en stelt op basis daarvan de factuur op.

Maar er geraken al eens papieren, e-mails verloren en sommige zaken worden vergeten door gebrek aan uniformiteit.

Er is een oplossing nodig.

Opdracht

De klant heeft weinig kaas gegeten van computers, maar beseft wel dat zijn manier van werken een stuk gemakkelijker zou kunnen gemaakt worden door het maken van een webapplicatie. Hij had zelf het volgende idee:

Een all-in-one webapplicatie waarbij hij de uren niet meer zelf hoeft in te vullen, maar reeds gebeurd is door het personeel / onderaannemers die hij uitsuurt.

Mobiele WebApp

Het personeel/onderaannemers zullen via hun smartphone via een progressive web app - na inloggen - dagelijks het volgende kunnen invullen:

klant

datum

startuur

einduur

pauze

uitgevoerde activiteiten

gebruikte materialen (vb schroeven, kabels, sensoren, ...)

transport (aantal kilometers)

Onderaannemers hebben ook een extra optie waarbij ze hun tarief kunnen wijzigen (uurtarief, transportkost)

Het zou handig zijn als de gebruikers van die app ook kunnen bekijken hoeveel uren ze in totaal al gepresteerd hebben (per week / per maand / per klant). Bedenk zelf een aantal praktische toevoegingen.

Backend

In de backend worden alle medewerkers van ARTE-TECH toegevoegd.

In de backend worden alle klanten van ARTE-TECH toegevoegd.

Het uurtarief van de techniekers en transportkost wordt per bedrijf op voorhand afgesproken en ingevoerd in de backend.

Volgende regel geldt altijd voor uurtarief bij uitzonderingen:

overuren tijdens de week (> 8 uur): 20% meer

zaterdag: 50% meerkost

zondag: dubbel tarief

De manager kan op een gebruiksvriendelijke manier de gepresteerde uren opvolgen, die doorgegeven zijn door zijn personeel/onderaannemers (en zo nodig ook corrigeren of handmatig invullen).

De backend zou een functie moeten hebben dat een periode kan opgesteld worden en kan gemeld worden naar de klant (het bedrijf) per mail.

Deze klant moet de periode nakijken via de website (zie hieronder) en bevestigen. (tot nog toe gebeurde dat immers op papier)

Via de backend moet het mogelijk zijn om totaalprijs per periode te berekenen en dit te exporteren naar een overzichtelijke xls / pdf.

Je hoeft geen rekening te houden met BTW (alles excl. btw)

Facturatie is niet nodig (maar mag uiteraard)

Website

Elke klant (de bedrijven dus) kan de gepresteerde uren nakijken via hun computer.

Zij volgen dit op per periode (na notificatie per mail), waarbij ze in detail alle prestaties kunnen nakijken en kunnen bevestigen.

Indien er iets zou mis zijn, hebben ze de mogelijkheid om een opmerking te versturen.

Er is ook een mogelijkheid om een periode als overzichtelijke pdf te printen voor eigen administratie.

Analyse

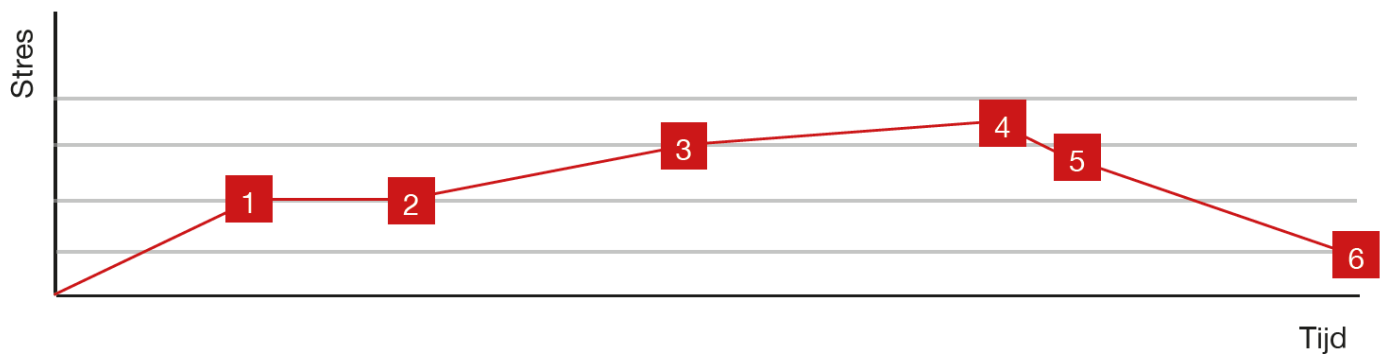
Scenario

De betrokken persoon is een client , medewerker of een eigenaar van het bedrijf ARTE-TECH. Deze app is eerder een volgsysteem voor iedereen. De admin kan alles van iedereen zien, de client kan taken zien en hiervan een overzicht in pdf of mail vorm ontvangen en een medewerker kan taken toevoegen en/of bewerken.

Doel en verwachtingen

- 1) Vootbrengen van een webapp
- 2) voortbrengen vann een admin systeem in symfony
- 3) Voldoen aan klanten eisen
- 4) Product moet af zijn op 05/01/20

Ontwerpen	Development	Dossier	Verdedeging
<ol style="list-style-type: none">1) Style guide2) Sitemap3) Wireframes4) Visual designs	<ol style="list-style-type: none">1) Database2) Front-/Back-end3) Content4) Lay-out5) Testing	<ol style="list-style-type: none">1) Verzamelen info2) Implementeren3) Lay-out	<ol style="list-style-type: none">1) Presentatie maken2) Presenteren



- 1) Aanmaken database model
- 2) Extra database tabellen zijn vereist
- 3) De token ophalen in de webapp met React wou niet werken
- 4) Pdf en mail functie blijven kleine problemen geven
- 5) Besef dat dossier moet gemaakt worden
- 6) Deadline is bereikt

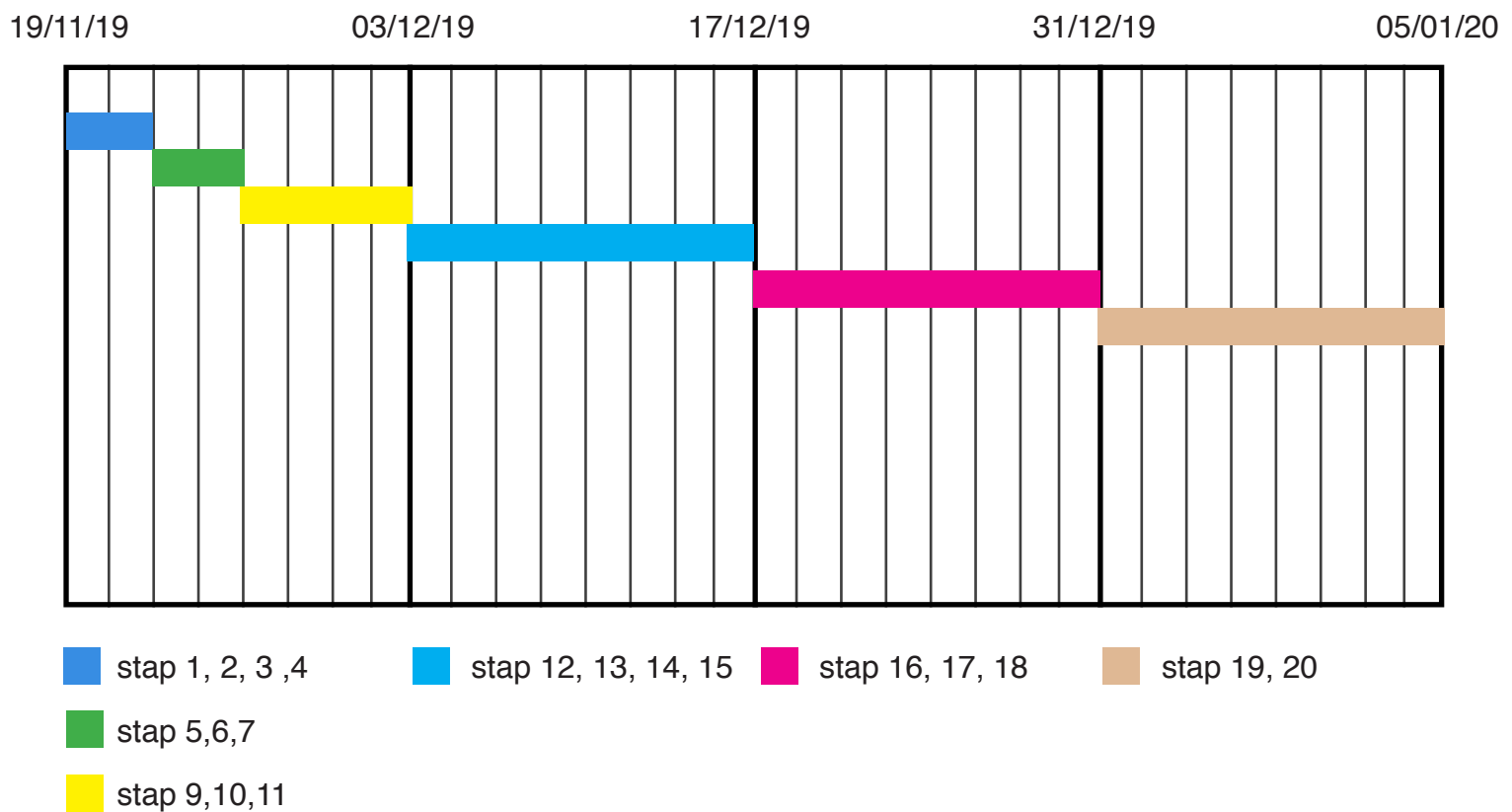
Opportunities

Symfony kennis opfrissen	Een nieuw project voor mijn portfolio	Vorbereiding voor BAP
--------------------------	---------------------------------------	-----------------------

Planning

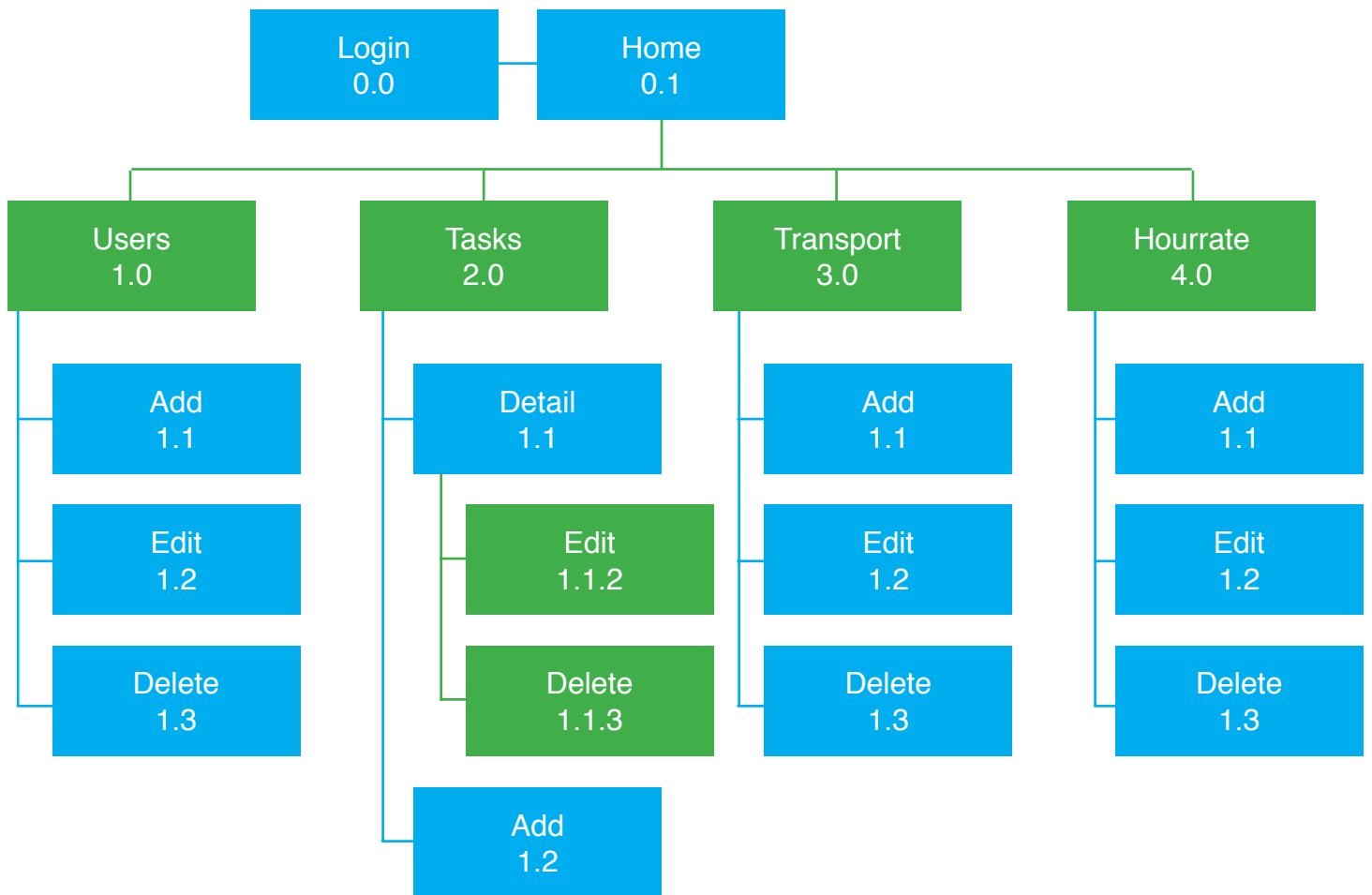
Prioriteren Activiteiten

- 1) Briefing bijwonen
- 2) Sitemap opstellen
- 3) Opzetten Project I Structuur van het project
- 4) Opzetten database
- 5) Aanmaken wireframes
- 6) Design maken
- 7) Testcontent toevoegen aan database
- 8) Login systeem opzetten I Admin invoegen
- 9) opzetten admin pagina's met check
- 10) admin functies toevoegen
- 11) afwerken admin systeem
- 12) Aanmaken react app
- 13) Login met token aanmaken
- 14) CRUD systeem opstellen
- 15) afwerken React app
- 16) Email / pdf toevoegen bij symfony
- 17) Klant gedeelte afwerken
- 18) Css op alles toevoegen
- 19) Productie dossier
- 20) Testing

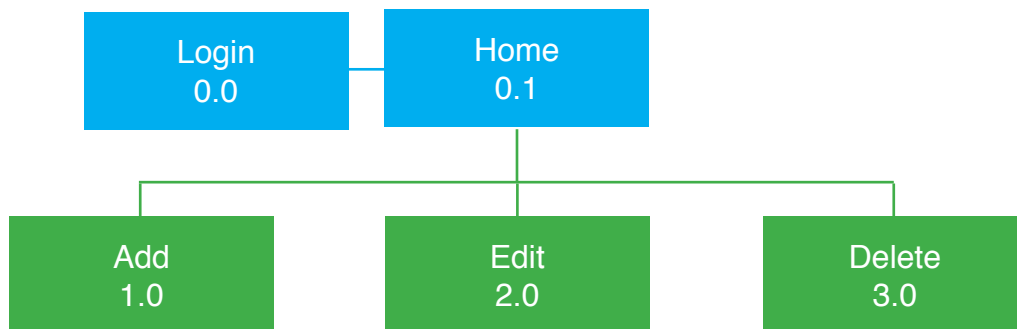


Sitemap

Website



Website



Wireframes

Link 1Link 2Link 3Link 4Link 5Link 6

Give the amount of days that you want the period to be

MAL

Give the amount of days that you want the period to be

PDF

Tasks

Task

Task

Task

See more

Link 1Link 2Link 3Link 4Link 5Link 6

Tasks

Task

Task

Task

Task

Task

Task

Link 1Link 2Link 3Link 4Link 5Link 6

Admin Dashboard

Actions

Link 1Link 2Link 3Link 4Link 5Link 6

Usersadd

Name	Email	Tel	Function	
Name	Email	Tel	Function	Edit Delete
Name	Email	Tel	Function	Edit Delete
Name	Email	Tel	Function	Edit Delete
Name	Email	Tel	Function	Edit Delete
Name	Email	Tel	Function	Edit Delete
Name	Email	Tel	Function	Edit Delete
Name	Email	Tel	Function	Edit Delete

Link 1Link 2Link 3Link 4Link 5Link 6

Edit User

Name

Email

Tel

Pass

Edit

Link 1Link 2Link 3Link 4Link 5Link 6

Add User

Name

Email

Tel

Pass

Add

Link 1Link 2Link 3Link 4Link 5Link 6

Tasksadd

Name	Email	Function	Tasks
Name	Email	Function	Tasks
Name	Email	Function	Tasks
Name	Email	Function	Tasks
Name	Email	Function	Tasks
Name	Email	Function	Tasks
Name	Email	Function	Tasks
Name	Email	Function	Tasks

Link 1Link 2Link 3Link 4Link 5Link 6

Name

Task

Task

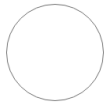
Task

Task

MAL

PDF

Wireflow



Add Task

Welkom
User

Hier kan u als medewerker van ARTE-TECH uw
taken per dag invullen en doorsturen



Tasklist

Edit Task

Designs

Link 1Link 2Link 3Link 4Link 5Link 6

Give the amount of days that you want the period to be

MAIL

Give the amount of days that you want the period to be

PDF

Tasks

Task

Task

Task

See more

Link 1Link 2Link 3Link 4Link 5Link 6

Tasks

Task

Task

Task

Task

Task

Task

Link 1Link 2Link 3Link 4Link 5Link 6

Admin Dashboard

Actions

Link 1Link 2Link 3Link 4Link 5Link 6

Users

add

Name	Email	Tel	Function	
Name	Email	Tel	Function	Edit Delete
Name	Email	Tel	Function	Edit Delete
Name	Email	Tel	Function	Edit Delete
Name	Email	Tel	Function	Edit Delete
Name	Email	Tel	Function	Edit Delete
Name	Email	Tel	Function	Edit Delete
Name	Email	Tel	Function	Edit Delete

Link 1Link 2Link 3Link 4Link 5Link 6

Edit User

Name

Name

Email

Email

Tel

Tel

Pass

Pass

Edit

Link 1Link 2Link 3Link 4Link 5Link 6

Add User

Name

Email

Tel

Pass

Add

Link 1Link 2Link 3Link 4Link 5Link 6

Tasks

add

Name	Email	Function	Tasks
Name	Email	Function	Tasks
Name	Email	Function	Tasks
Name	Email	Function	Tasks
Name	Email	Function	Tasks
Name	Email	Function	Tasks
Name	Email	Function	Tasks
Name	Email	Function	Tasks

Link 1Link 2Link 3Link 4Link 5Link 6

Name

Task

Task

Task

Task

Give the amount of days that you want the period to be

MAIL

Give the amount of days that you want the period to be

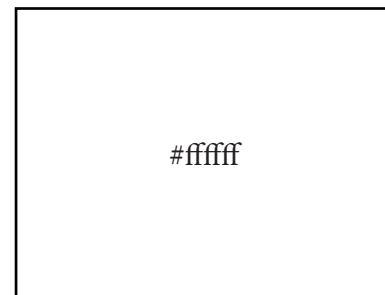
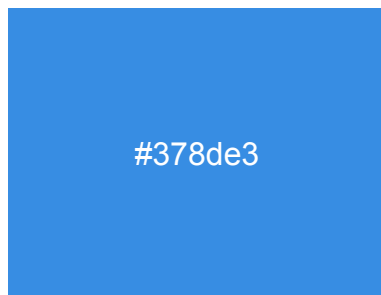
PDF

Owen De Waele | NMD 2019-20

17

Style guide

Main colors



Text

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Arial regular 12p

Header 1

Arial regular 24p

Header 2

Arial regular 18p

Buttons



Edit



Delete



Add

Code snippets

```

.../**
... * @Route("/mail/{days}/{id}", name="mail", methods={"GET"})
... */
... public function mail($days, $id, MailerInterface $mailer, TokenStorageInterface $tokenStorage)
... {
...     $usr = $tokenStorage->getToken()->getUser();
...     if($usr == 'anon.'){
...         return $this->redirect('/login');
...     }else{
...         if($usr->getPosition() == 'Admin' || $usr->getPosition() == 'User'){
...             //
...         }else{
...             return $this->redirect('/');
...         }
...     }
...     $manager = $this->getDoctrine()->getManager();
...     $owner = $manager->getRepository(User::class)->findBy(
...         ['Function' => 'Owner'],
...     );
...     $currentuser = $manager->getRepository(User::class)->find($id);
...     $now = new \DateTime();
...     $thirtyDaysAgo = $now->sub(new \DateInterval("P".$days."D"));
...     $qb = $manager->createQueryBuilder();
...     if($currentuser->getFunction() == "Client"){
...         $qb->select('e')
...         ->from(Tasks::class, 'e')
...         ->where('e.Client_id = '.$currentuser->getId())
...         ->andWhere($qb->expr()->between(
...             'e.Start',
...             ':from',
...             ':to'
...         ))
...         ->setParameters(array('from' => $thirtyDaysAgo, 'to' => new \DateTime()));
...
...     $tasks = $qb->getQuery()->getResult();
...     $prices = [];
...     foreach($tasks as $task){
...         $taskuser = $manager->getRepository(User::class)->find($task->getEmployeeId());
...         $day = $task->getCurrentDay();
...         $timeworked = $task->getDatediff();
...         $traveldistance = $task->getTraveldistance();
...         $hourrate = $taskuser->getHourrate();
...         $travelrate = $taskuser->getTransportrate();
...
...         if($day == "Sun"){
...             $uren = $timeworked*$hourrate*2;
...             $afstand = $traveldistance * $traveldistance;
...             $price = $uren + $afstand;
...         }
...         else if($day == "Sat"){
...             $price = $timeworked*$hourrate*1.5 + ($traveldistance * $traveldistance);
...         }else{
...             if($timeworked > 8){
...                 $overhours = ($timeworked - 8) * ($hourrate*1.2);
...                 $normalhours = 8 * $hourrate;
...                 $price = ($overhours + $normalhours) + ($traveldistance * $traveldistance);
...             }else{
...                 $price = ($timeworked * $hourrate) + ($traveldistance * $traveldistance);
...             }
...         }
...         array_push($prices, ["id" => $task->getId(), "price" => $price]);
...     }
...     $message = (new TemplatedEmail())
...     ->from("abustering841@gmail.com")
...     ->to($currentuser->getEmail())
...     ->htmlTemplate('functions/email.html.twig')
...     ->context([
...         'controller_name' => 'FunctionController',
...         'tasks' => $tasks,
...         'User' => $currentuser->getName(),
...         'prices' => $prices
...     ]);
...     $mailer->send($message);
...     return $this->redirect('/tasks/'.$id);

```

```

    /**
     * @Route("/tasks/{id}", name="user-tasks")
     */
    public function detail($id, TokenStorageInterface $tokenStorage, Request $request)
    {
        $usr = $tokenStorage->getToken()->getUser();
        if($usr == 'anon.'){
            return $this->redirect('/login');
        }else{
            if($usr->getPosition() == 'Admin'){
                // ...
            }else{
                return $this->redirect('/');
            }
        }
        $defaultData = ['message' => 'Period form'];
        $form = $this->createFormBuilder($defaultData)
            ->add('Days', TextType::class)
            ->add('send', SubmitType::class)
            ->getForm();
        if ($request->getMethod() == 'POST') {
            // data is an array with "name", "email", and "message" keys
            $data = $request->request->get('form');
            return $this->redirect('/mail/'.$data['Days'].'/'.$id);
        }
        $tasks = $this->getDoctrine()->getRepository(Tasks::class)->findBy(
            ['Employee_id' => $id]
        );
        return $this->render('task/detail.html.twig', [
            'controller_name' => 'TaskController',
            'tasks' => $tasks,
            'form' => $form->createView(),
        ]);
    }

```

```

    /**
     * @Route("/delete-task/{id}", name="delete-task")
     */
    public function delete($id, TokenStorageInterface $tokenStorage)
    {
        $usr = $tokenStorage->getToken()->getUser();
        if($usr == 'anon.'){
            return $this->redirect('/login');
        }else{
            if($usr->getPosition() == 'Admin'){
                // ...
            }else{
                return $this->redirect('/');
            }
        }
        $manager = $this->getDoctrine()->getManager();
        $task = $manager->getRepository(Tasks::class)->find($id);
        $manager->remove($task);
        $manager->flush();
        return $this->redirectToRoute('tasks');
    }
    /**
     * @Route("/tasks", name="tasks")
     */
    public function tasks(TokenStorageInterface $tokenStorage )
    {
        $usr = $tokenStorage->getToken()->getUser();
        if($usr == 'anon.'){
            return $this->redirect('/login');
        }else{
            if($usr->getPosition() == 'Admin'){
                // ...
            }else{
                return $this->redirect('/');
            }
        }
        $users = $this->getDoctrine()->getRepository(User::class)->findAll();
        return $this->render('task/index.html.twig', [
            'controller_name' => 'TaskController',
            'users' => $users,
        ]);
    }

```

```

    $task = New Tasks();
    $form = $this->createFormBuilder($task)
        ->add('EmployeeId', EntityType::class, [
            'label' => 'Employee',
            'class' => User::class,
            'query_builder' => function(EntityRepository $er){
                return $er->createQueryBuilder('u')
                    ->where('u.Function =:employee OR u.Function =:freelance')
                    ->setParameters(array('employee'=>'Employee', 'freelance'=>'Freelance'));
            },
            'choice_label' => function(User $user){
                return $user->getId().' '.$user->getName();
            },
        ])
        ->add('ClientId', EntityType::class, [
            'label' => 'Client',
            'class' => User::class,
            'query_builder' => function(EntityRepository $er){
                return $er->createQueryBuilder('u')
                    ->where('u.Function =:klant')
                    ->setParameter('klant','Client');
            },
            'choice_label' => function(User $user){
                return $user->getId().' '.$user->getName();
            },
        ])
        ->add('Start', DateTimeType::class, [
            'label' => 'Start',
            'date_widget' => 'choice',
            'time_widget' => 'choice',
        ])
    ]

```

```

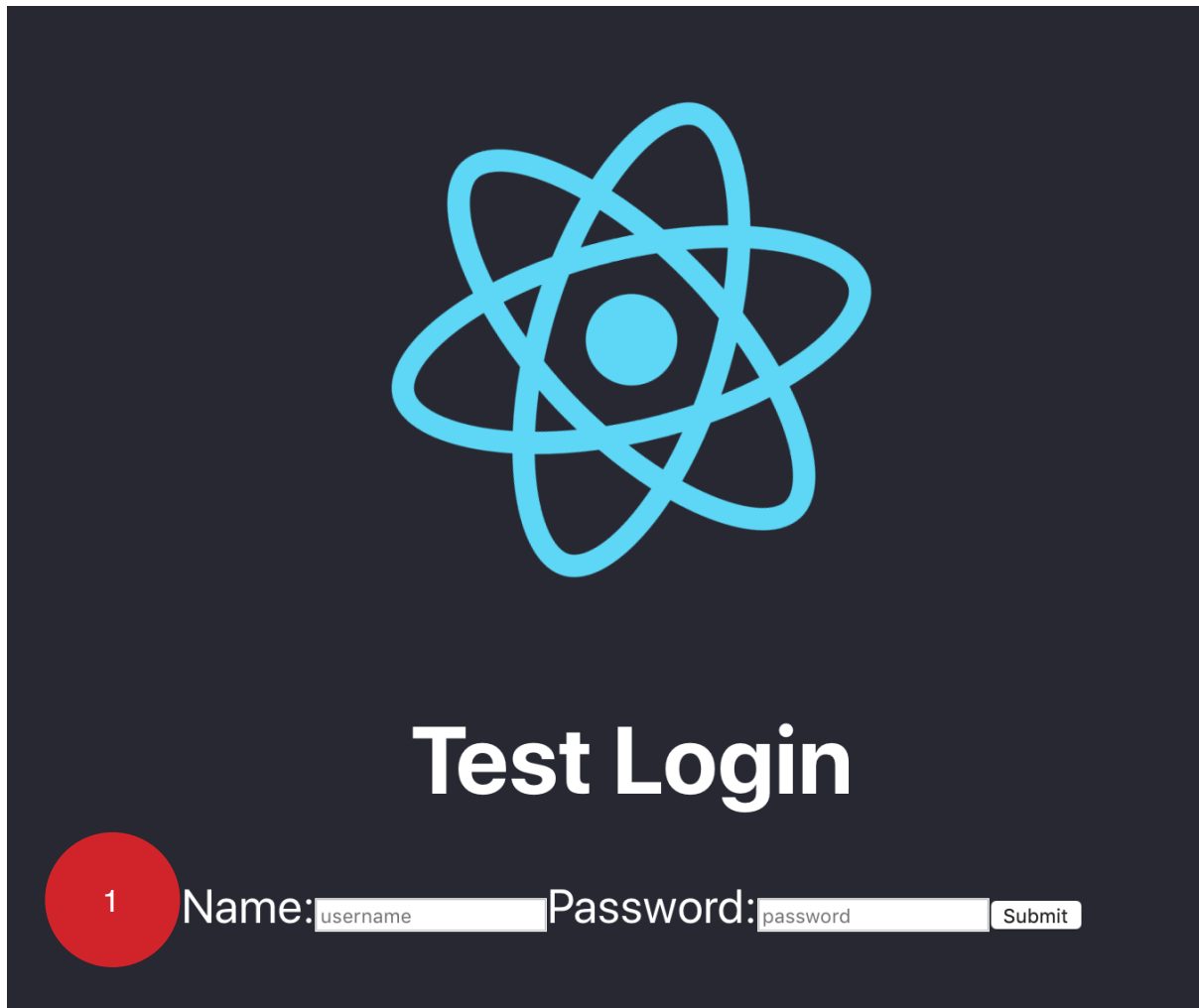
$pdfOptions = new Options();
$pdfOptions->set('defaultFont', 'Arial');
$dmpdf = new Dompdf($pdfOptions);
$html = $this->renderView('functions/pdf.html.twig', [
    'title' => "Welcome to our PDF Test",
    'controller_name' => 'functioncontroller',
    'tasks' => $tasks,
    'User' => $currentuser->getName(),
    'prices' => $prices
]);
$dmpdf->loadHtml($html);
$dmpdf->setPaper('A4', 'portrait');
$dmpdf->render();
$dmpdf->stream("mypdf.pdf", [
    "Attachment" => false
]);
return new Response('response');

```

Userguide

React app

Note styled screens will follow



To start of when you enter the react app you will be guided to a login screen where your credentials will be checked with the ones in the database so use the 1) login form and you will be guided to your home page.

TaskList

11

Voor klant: 9

Activiteiten: niet veel

Materiaal: ook niet zo veel

Aantal km: 10

Gewerkte uren: 2

3

Delete

Edit

2

Na te hebben ingelogd zal u uw homescreen zien met daar al uw taken opgelijst met de optie om te 2) editen en te 3) deleten je hebt ook de optie uiteraard om een task toe te voegen zoals bij de andere foto te zien is 4).

Logout

Client: Start: End: Break: Activities: Materials: Traveldistance:

Submit

4

Symfony app

Note styled screens will follow

Please sign in

Name Password

Om te starten zal als je u wilt aanmelden het 1) login formulier moeten gebruiken dit is uiteraard enkel voor de klant en de admin. Na het inloggen van de admin gaan we naar het user overzicht, hier kunnen we 2) een user registreren 3) editen en 4) deleten.

- [Users](#)
- [Tasks](#)
- [Hourrates](#)
- [Transportrates](#)
- [Home](#)

2

[Add](#)

Name	Email	Tel	Function	
Texaco	texaco@texaco.be	23332423432		Edit Delete
Shell	Shell@Shell.be	092342309	Client	Edit Delete
werknemer1	owendewa@student.arteveldhehs.be	02934	Employee	Edit Delete
werknemer2	werknemer2@werknemer2.be	02934	Employee	Edit Delete
werknemer3	werknemer3@werknemer3.be	02934	Employee	Edit Delete
freelance1	freelance1@freelance1.be	02934	Freelance	Edit Delete
freelance2	freelance2@freelance2.be	02934	Freelance	Edit Delete
freelance3	freelance3@freelance3.be	02934	Freelance	Edit Delete
Owen De Waele	owen.de.waele@hotmail.com	0471031395	Owner	Edit Delete

Deze structuur wordt ook gehanteerd bij de hourrates en transportrates. Nu om over te gaan op 5) het registerform waar na het toevoegen van een persoon deze in de database terecht komt. Daarom is er de optie in dit formulier om de 6) rol van de user aan te duiden.

Nu voor de uitzondering te bespreken als we naar het overzicht gaan van de tasks zal u zien dat er geen edit of delete staat maar 7) tasks als u hierop verder gaat zal u zien dat iedere user zowel klant als werknemer een eigen details page heeft met zijn/haar taak hierop kan u een 8) mail sturen naar de betrokken persoon en/of een pdf downloaden met hun tasks

- [Users](#)
- [Tasks](#)
- [Hourrates](#)
- [Transportrates](#)
- [Home](#)

Register

5

Name

Plain password

Tel

Email

Function

☐ Employee ☐ Freelance ☐ Owner ☐ Client

Agree terms ☐

6

[Add](#)

Name	Email	Function
Owen De Waele	owen.de.waele@hotmail.com	Owner

7

[Tasks](#)

- [Users](#)
- [Tasks](#)
- [Hourrates](#)
- [Transportrates](#)
- [Home](#)

Days

8

Employee: 10

Client: 9

From: 2019-12-29 12:00:00

Until: 2019-12-29 14:00:00

Break Time: 0

Activities done: niet veel

Materials used: ook niet zo veel

Traveldistance: 10km

Dag: Sun

- - -

Deploymentguide

Om de Deployment Guide te starten, moet u naar <https://github.com/gdm-1718-owendewa/Cmsdev> en download het zipbestand of kloon het naar je github-bureaublad. U kunt de werkgelegenheidsgids volgen in het Leesmij-bestand of hieronder

Nadat je de map hebt gedownload of gekloond, ga je naar de hoofdmap in ons geval eindwerk_cmsdev en installeer je composert lokaal of global als je geen composer hebt geïnstalleerd ga naar <https://getcomposer.org/download/> en volg de stappen. of als u een snelle installatie in de huidige map (bij voorkeur de hoofdmap) wilt uitvoeren.

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"  
php -r "if (hash_file('sha384', 'composer-setup.php') === '48e3236262b34d-  
30969dca3c37281b3b4bbe3221bda826ac6a9a62d6444cdb0dcd0615698a5cbe587c3f-  
0fe57a54d8f5')  
{ echo 'Installer verified'; } else { echo 'Installer corrupt';  
unlink('composer-setup.php'); } echo PHP_EOL;"  
php composer-setup.php  
php -r "unlink('composer-setup.php');"
```

Nadat u composert hebt geïnstalleerd, moet u het .env-bestand maken en de stappen volgen die hier worden getoond <https://medium.com/@joeymasip/how-to-create-an-api-with-symfony-4-and-JWT-b2334a8fbec2> om je eigen JWT-tokens te krijgen.

Ga nu naar het .env-veld en voeg toe
MAILER_DSN = gmail: // {{email}}: {{wachtwoord}} @default om de gmail-services te laten werken.

Nu kunt u de symfony-app uitvoeren met symfony-server uitgevoerd
Er zijn geen verdere acties vereist.

Nu kunnen we naar onze webapp-map gaan en een yarn installatie uitvoeren.
Hierna moet je een yarn server:start commando in de terminal en het zou goed moeten zijn om te gaan.
Vergeet niet om altijd te controleren of de symfony-app actief is voordat u de CRUD gebruikt functies van de react-app